In the Claims:

Please amend the claims as indicated below:

1. (Currently amended) A system, comprising:

a client comprising a client Web services stack that supports both a markup language protocol and a binary encoding protocol; and

a server comprising a server Web services stack that supports both the markup language protocol and the binary encoding protocol, wherein the server Web services stack is configured to:

communicate with the client Web services stack according to the markup language protocol; and

dynamically switch to communicate with the client Web services stack according to the binary encoding protocol;

wherein the client Web services stack and the server Web services stack each support the markup language protocol and the binary encoding protocol with a single API (application programming interface).

2. (Currently amended) The system as recited in claim 1, wherein the client is a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) client.

3. (Currently amended) The system as recited in claim 1, wherein the client is a J2ME (Java 2 Micro Edition) client.

4. (Currently amended) The system as recited in claim 1, wherein the server is a

JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server.

5. (Currently amended) The system as recited in claim 1, wherein the markup language protocol is XML (eXtensible Markup Language).

6. (Currently amended) The system as recited in claim 1, wherein communication according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from a markup language schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

7. (Original) The system as recited in claim 1, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

8. (Original) The system as recited in claim 1, further comprising another client comprising another client Web services stack that supports only the binary encoding protocol, and wherein the server Web services stack is further configured to communicate with the other client Web services stack according to the binary encoding protocol.

9. (Original) The system as recited in claim 1, wherein, to communicate with the client Web services stack according to the binary encoding protocol, the server Web services stack is further configured to:

translate the markup language protocol to binary encoding protocol messages for transmission to the client Web services stack; and

translate binary encoding protocol messages received from the client Web services stack to the markup language protocol.

10. (Currently amended) The system as recited in claim 1, wherein, to communicate with the client Web services stack according to the binary encoding protocol, the server

Web services stack is further configured to serialize the markup language protocol to generate binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

11. (Currently amended) The system as recited in claim 1, wherein, to communicate with the client Web services stack according to the binary encoding protocol, the server Web services stack is further configured to serialize the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for transmitting data described by markup language schema.

12. (Currently amended) A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a Web services stack configured to:

communicate with another Web services stack on another system according to the markup language protocol; and

dynamically switch to communicate with the other Web services stack according to the binary encoding protocol;

wherein the Web services stack supports the markup language protocol and the binary encoding protocol with a single API (application programming interface).

13. (Currently amended) The system as recited in claim 12, wherein the system is a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote

Procedure Call)) client.

14. (Currently amended) The system as recited in claim 12, wherein the system is a J2ME (Java 2 Micro Edition) client.

15. (Currently amended) The system as recited in claim 12, wherein the system is a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server.

16. (Original) The system as recited in claim 12, wherein the system and the other system are peers on a network.

17. (Currently amended) The system as recited in claim 12, wherein the markup language protocol is XML (eXtensible Markup Language).

18. (Currently amended) The system as recited in claim 12, wherein communication according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from a markup language schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

19. (Original) The system as recited in claim 12, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

20. (Original) The system as recited in claim 12, wherein, to communicate with the other Web services stack according to the binary encoding protocol, the Web services stack is further configured to:

translate the markup language protocol to binary encoding protocol messages for transmission to the other Web services stack; and

translate binary encoding protocol messages received from the other Web services

stack to the markup language protocol.

21. (Currently amended) The system as recited in claim 12, wherein, to communicate with the other Web services stack according to the binary encoding protocol, the Web services stack is further configured to serialize the markup language protocol to generate binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

22. (Currently amended) The system as recited in claim 12, wherein, to communicate with the other Web services stack according to the binary encoding protocol, the Web services stack is further configured to serialize the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for transmitting data described by markup language schema.

23. (Currently amended) A system, comprising:

means for communicating between a Web services stack on the system and another Web services stack on another system according to a markup language protocol; and

means for dynamically switching to communicate between the Web services stack and the other Web services stack according to a binary encoding protocol;

wherein the Web services stack supports the markup language protocol and the binary encoding protocol with a single API (application programming interface).

24. (Currently amended) The system as recited in claim 23, wherein the markup language protocol is XML (eXtensible Markup Language), and wherein communication according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from an XML

(eXtensible Markup Language) schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

25. (Original) The system as recited in claim 24, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

26. (Currently amended) A method, comprising:

a Web services stack communicating with another Web services stack according to a markup language protocol; and

the Web services stack dynamically switching to communicating with the other Web services stack according to a binary encoding protocol;

wherein the Web services stack supports the markup language protocol and the binary encoding protocol with a single API (application programming interface).

27. (Currently amended) The method as recited in claim 26, wherein the Web services stack is implemented on a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) client system.

28. (Currently amended) The method as recited in claim 26, wherein the Web services stack is implemented on a J2ME (Java 2 Micro Edition) client system.

29. (Currently amended) The method as recited in claim 26, wherein the Web services stack is implemented on a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server system.

30. (Original) The method as recited in claim 26, wherein the Web services stack and the other Web services stack are implemented on peers on a network.

31. (Currently amended) The method as recited in claim 26, wherein the markup language protocol is XML (eXtensible Markup Language).

32. (Currently amended) The method as recited in claim 26, wherein communicating according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from a markup language schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

33. (Original) The method as recited in claim 26, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

34. (Original) The method as recited in claim 26, wherein said communicating with the other Web services stack according to the binary encoding protocol comprises:

translating the markup language protocol to binary encoding protocol messages for transmission to the other Web services stack; and

translating binary encoding protocol messages received from the other Web services stack to the markup language protocol.

35. (Currently amended) The method as recited in claim 26, wherein said communicating with the other Web services stack according to the binary encoding protocol comprises serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

36. (Currently amended) The method as recited in claim 26, wherein said communicating with the other Web services stack according to the binary encoding protocol comprises serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for

transmitting data described by markup language schema.

37. (Currently amended) A computer-accessible storage medium comprising program instructions, wherein the program instructions are configured to implement:

a Web services stack communicating with another Web services stack according to a markup language protocol; and

the Web services stack dynamically switching to communicating with the other Web services stack according to a binary encoding protocol;

wherein the Web services stack supports the markup language protocol and the binary encoding protocol with a single API (application programming interface).

38. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein the Web services stack is implemented on a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) client system.

39. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein the Web services stack is implemented on a J2ME (Java 2 Micro Edition) client system.

40. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein the Web services stack is implemented on a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server system.

41. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein the Web services stack and the other Web services stack are

implemented on peers on a network.

42. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein the markup language protocol is XML (eXtensible Markup Language).

43. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein communicating according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from a markup language schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

44. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

45. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein, in said communicating with the other Web services stack according to the binary encoding protocol, the program instructions are further configured to implement:

translating the markup language protocol to binary encoding protocol messages for transmission to the other Web services stack; and

translating binary encoding protocol messages received from the other Web services stack to the markup language protocol.

46. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein, in said communicating with the other Web services stack according to the binary encoding protocol, the program instructions are further configured to implement serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

47. (Currently amended) The computer-accessible storage medium as recited in claim 37, wherein, in said communicating with the other Web services stack according to the binary encoding protocol, the program instructions are further configured to implement serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for transmitting data described by markup language schema.

48. (Currently amended) A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a Web services stack configured to:

communicate with other systems using either a binary encoding protocol or a markup language protocol using a single API (application programming interface);

negotiate with another system to determine if the other system supports the binary encoding protocol;

if the other system supports the binary encoding protocol, communicate with the other system according to the binary encoding protocol; and

if the other system does not support the binary encoding protocol, communicate with the other system according to the markup language protocol.

49. (Currently amended) The system as recited in claim 48, wherein the system is a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server.

50. (Original) The system as recited in claim 48, wherein the system and the other system are peers on a network.

51. (Currently amended) The system as recited in claim 48, wherein the markup language protocol is XML (eXtensible Markup Language).

52. (Currently amended) The system as recited in claim 48, wherein communication according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from a markup language schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

53. (Original) The system as recited in claim 48, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

54. (Original) The system as recited in claim 48, wherein the Web services stack is further configured to, if the other system includes a Web services stack configured to communicate with either the binary encoding protocol or the markup language protocol:

communicate with the other system according to the markup language protocol; and

dynamically switch to communicate with the other system according to the binary encoding protocol.

55. (Currently amended) The system as recited in claim 48, wherein, to communicate with the other system according to the binary encoding protocol, the Web services stack is further configured to serialize the markup language protocol to generate

binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

56. (Currently amended) The system as recited in claim 48, wherein, to communicate with the other system according to the binary encoding protocol, the Web services stack is further configured to serialize the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for transmitting data described by markup language schema.

57. (Currently amended) A system, comprising:

means for communicating with other systems using either a binary encoding protocol or a markup language protocol using a single API (application programming interface);

means for negotiating with another system to determine if the other system supports the binary encoding protocol; and

means for communicating with the other system according to the binary encoding protocol if the other system supports the binary encoding protocol.

58. (Currently amended) The system as recited in claim 57, wherein the markup language protocol is XML (eXtensible Markup Language), and wherein communication according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from an XML (eXtensible Markup Language) schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

59. (Original) The system as recited in claim 57, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

60. (Currently amended) A method, comprising:

a Web services stack on a system configured to communicate with other systems using either a binary encoding protocol or a markup language protocol using a single API (application programming interface) negotiating with another system to determine if the other system supports the binary encoding protocol;

if the other system supports the binary encoding protocol, the Web services stack communicating with the other system according to the binary encoding protocol; and

if the other system does not support the binary encoding protocol, the Web services stack communicating with the other system according to the markup language protocol.

61. (Currently amended) The method as recited in claim 60, wherein the system is a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server.

62. (Original) The method as recited in claim 60, wherein the system and the other system are peers on a network.

63. (Currently amended) The method as recited in claim 60, wherein the markup language protocol is XML (eXtensible Markup Language), and wherein communicating according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from an XML (eXtensible Markup Language) schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

64. (Original) The method as recited in claim 60, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

65. (Original) The method as recited in claim 60, further comprising, if the other system includes a Web services stack configured to communicate with either the binary encoding protocol or the markup language protocol:

communicating with the other system according to the markup language protocol; and

dynamically switching to communicating with the other system according to the binary encoding protocol.

66. (Currently amended) The method as recited in claim 60, wherein said communicating with the other system according to the binary encoding protocol comprises serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

67. (Currently amended) The method as recited in claim 60, wherein said communicating with the other system according to the binary encoding protocol comprises serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for transmitting data described by markup language schema.

68. (Currently amended) A computer-accessible storage medium comprising program instructions, wherein the program instructions are configured to implement:

a Web services stack on a system configured to communicate with other systems using either a binary encoding protocol or a markup language protocol using a single API (application programming interface) negotiating with another system to determine if the other system supports the binary encoding protocol;

if the other system supports the binary encoding protocol, the Web services stack communicating with the other system according to the binary encoding protocol; and

if the other system does not support the binary encoding protocol, the Web services stack communicating with the other system according to the markup language protocol.

69. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein the system is a JAX-RPC (Java API for XML (eXtensible Markup Language)-based RPC (Remote Procedure Call)) server.

70. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein the system and the other system are peers on a network.

71. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein the markup language protocol is XML (eXtensible Markup Language), and wherein communicating according to the binary encoding protocol ~~is WS-Fast~~ comprises mapping from an XML (eXtensible Markup Language) schema to an ASN.1 (Abstract Syntax Notation One) schema, and generating a binary encoding from the ASN.1 schema.

72. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein the binary encoding protocol uses Packed Encoding Rules (PER) encoding.

73. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein the program instructions are further configured to implement, if the other system includes a Web services stack configured to communicate with either the binary encoding protocol or the markup language protocol:

communicating with the other system according to the markup language protocol; and

dynamically switching to communicating with the other system according to the binary encoding protocol.

74. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein, in said communicating with the other system according to the binary encoding protocol, the program instructions are further configured to implement serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Infoset~~ a self-describing binary format that preserves the markup language protocol information set.

75. (Currently amended) The computer-accessible storage medium as recited in claim 68, wherein, in said communicating with the other system according to the binary encoding protocol, the program instructions are further configured to implement serializing the markup language protocol to generate binary encoding protocol messages according to ~~Fast Schema~~ a schema-optimized binary format for transmitting data described by markup language schema.